

## Übung 8: CSS

Abgabedatum: 12 Dez. 2019, 12:00 Uhr


### Aufgabe 1: Responsive Web Design

2+2=4 Punkte

- Unsere Webanwendung soll auch auf mobilen Systemen mit kleineren Bildschirmen verwendbar sein. Verändern Sie im ersten Schritt - so noch nicht geschehen, Ihr Layout so, dass es bei Vergrößern oder Verkleinern des Browsers die Breiten der unterschiedlichen Boxen skaliert (z.B. Flexbox oder prozentuale Breitenangaben). (2 P)
- Definieren Sie über *media-queries* eine Bildschirmbreite, unter der das Layout *umbricht*, in unserem Fall also die beiden Ansichten nicht mehr nebeneinander, sondern untereinander anzeigt. Betrachten Sie dazu die untenstehenden Screenshots und erweitern Sie Ihren CSS Code entsprechend. (2 P)

### Web Engineering Haltestellenübersicht

Home Stops



Ost
Ulm Ostplatz <input type="checkbox"/>
Wiblingen Ostermahnweg <input type="checkbox"/>
Ulm Ostpreußenweg <input type="checkbox"/>
Ulm ZOB Ost <input type="checkbox"/>

Copyright WebEng 2018, Maybe rights are reserved

### Web Engineering Routenplaner

Home Stops

**Favorit: Ulm Safranberg**

15:55 Linie 4 (Ulm Egertweg)  
15:58 Linie 4 (Grimmelfingen Eisenbahnstraße)  
16:05 Linie 4 (Ulm Egertweg)  
16:08 Linie 4 (Grimmelfingen Eisenbahnstraße)  
16:15 Linie 4 (Ulm Egertweg)

Ulm Universität Süd
Ulm Justizgebäude

Linie 13 (Ulm Kuhberg Schulzentrum): Ulm Universität Süd (15:54) -> Ulm Lupferbrücke (15:59)  
Linie 3 (Wiblingen Alte Siedlung): Ulm Lupferbrücke (15:59) -> Ulm Hauptbahnhof (16:15)  
Linie 1 (Ulm Ostpreußenweg): Ulm Hauptbahnhof (16:15) -> Ulm Justizgebäude (16:16)

Copyright WebEng 2018, Maybe rights are reserved

### Aufgabe 2: Blockworld - Positionierung mit CSS

1+2+2+2+1+2+2=12 Punkte

Laden Sie sich zunächst aus dem Moodle die Datei `material-blockworld.zip` herunter und entpacken Sie sie in ein Verzeichnis Ihrer Wahl. Enthalten ist ein HTML Dokument und eine CSS Datei. Modifizieren Sie für die folgenden Teilaufgaben nur das CSS Stylesheet. Betrachten Sie aber zunächst den Inhalt des HTML Dokuments sowohl im Browser als auch in einem Texteditor Ihrer Wahl (z.B. bietet dich der Atom Editor<sup>1</sup> für die Webentwicklung an). Stellen Sie die folgenden Angaben nach und nach her, verwenden Sie dazu die `position` CSS Eigenschaft, sowie damit zusammenhängende Eigenschaften. In manchen Teilaufgaben sind zusätzliche Eigenschaften (z.B. Farbe) zu

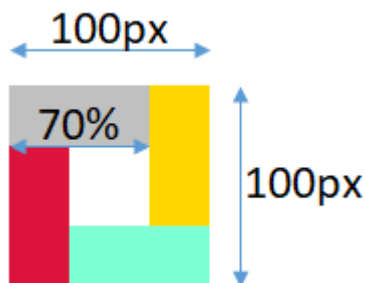
<sup>1</sup><https://atom.io>

nutzen. In vielen Fällen werden andere Teile des Layouts vorübergehend nicht mehr schön aussehen, darauf wird meist in den nächsten Teilaufgaben eingegangen.

- a) Das `main` Element soll im Browserfenster zentriert werden. Benutzen Sie hierfür die `margin` Eigenschaft. (1 P)
- b) Im HTML tauchen wiederholt Elemente der folgenden Struktur auf:

```
<div class="minibox">
  <div class="box"></div>
  <div class="box"></div>
  <div class="box"></div>
  <div class="box"></div>
</div>
```

Wir nennen diese im Folgenden `minibox` Elemente. Stellen Sie mittels CSS für die Klasse `minibox` und die vier Kind-`div` Elemente (Hinweis: Es gibt einen CSS Selektor, mit dem das n-te Kind eines Knotens ausgewählt werden kann) folgende Struktur her:



Die genauen Farben der vier Teilboxen sind dabei Ihnen überlassen, sie sollten aber gut unterscheidbar sein. Verwenden Sie für die Kind-`div` Elemente absolute Positionierung und erklären Sie in einem Kommentar, warum Sie dafür auch die `position` Eigenschaft des `minibox` Elements anpassen müssen. (2 P)

- c) Im `header` Element befinden sich zwei `Minibox` Elemente. Das erste soll am linken oberen Rand des Browsers und das zweite am rechten oberen Rand angezeigt werden. Sie rahmen also die Überschrift ein. *Hinweis: An dieser Stelle werden die beiden `Minibox` Elemente nicht mehr ganz im grün hinterlegten Bereich sein. Da wir uns nur mit Positionierung beschäftigen (und noch weitere Angaben folgen), ist dies an dieser Stelle zu vernachlässigen.* (2 P)
- d) Die Kopfzeile soll beim Herunterscrollen <sup>a</sup> immer am oberen Browser Rand sichtbar sein. (2 P)

<sup>a</sup>verkleinern sie ggf. die Höhe ihres Browserfensters oder erhöhen sie die `height` Angabe des `main` Elements

- e) Da die Kopfzeile nun aus dem normalen Layout Verfahren herausgerissen ist, beginnt in der anfänglichen Ansicht das `main` Element verdeckt (erkennbar am schwarzen Border des Elements). Ändern Sie die Positionierung des `main` Elements, dass es um 100px nach unten verschoben wird. (1 P)
- f) Beim Scrollen werden Sie nun feststellen, dass die Kopfzeile nun unter das `main` Element geschoben wird. Konsultieren Sie die Standardwerte für die Eigenschaft `position` sowie deren Effekte auf die Anordnung der HTML Elemente in Bezug auf ihre z-Koordinate.

Ändern Sie die z-Koordinate der Kopfzeile, sodass sie sich wieder vor das `main` Element beim Scrollen schiebt. (2 P)

- g) Unter dem Text im `main` Element befindet sich ein Element der Klasse `boxbar`. Stylen Sie das Element so, dass es erst nach 500px beginnt. Stylen Sie die im `boxbar` Element enthaltenen `minibox` Elemente so, dass sie beim Scrollen zunächst mitscrollen, aber schließlich einen Abstand von 200px zum oberen Rand nicht unterschreiten. (2 P)

## Aufgabe 3: Positionierung mit CSS3 Flexbox

5+2+2=9 Punkte

In CSS 3 wurden zwei neue Positionierungsoptionen hinzugefügt: Flexbox und Grid Layout. Aufgrund der Komplexität werden wir hier nur Flexbox diskutieren können, welches zum Beispiel auch für das bekannte CSS Framework Bootstrap <https://getbootstrap.com/> verwendet wird.

Verschaffen Sie sich zum Beispiel über <https://css-tricks.com/snippets/css/a-guide-to-flexbox/> noch einmal einen groben Überblick über die von Flexbox benutzten CSS Eigenschaften und ihre Auswirkungen.

Laden Sie aus dem Moodle das Archiv `material-flexbox.zip` herunter und entpacken Sie es. Enthalten sind wieder eine Website (`index.html`) und ein Stylesheet (`style.css`), welche gemeinsam ein paar Karteikarten anzeigen. Weiterhin enthalten ist ein einfaches Skript (`script.js`), welches die beiden Buttons im Header der Website mit Funktionalität versieht.

Experimentieren Sie ggf. mit verschiedenen Browserfenstergrößen, um das Verhalten Ihres Layouts zu testen.

a) Ignorieren Sie zunächst die Buttons. Stylen Sie das `ul` Element mit der id `flexi` als einen Flexbox Container und fügen Sie sowohl zum Container als auch zu den direkten Kindern entsprechend Flexbox Eigenschaften hinzu, um folgenden Sachverhalt herzustellen:

- Die Karten werden reihenweise angezeigt, wenn also nicht genug Platz in der aktuellen Reihe ist, soll die nächste Karte eine neue Reihe beginnen
- Die Breite der Karten soll sich an den zur Verfügung stehenden Platz anpassen, dabei soll keine Karte unter 200px breit angezeigt werden, Karten mit der Klasse `important` sollen doppelt so stark wachsen wie normale Karten
- Wenn Karten verschieden hoch sind, sollen sie entlang der Cross-Linie (siehe Definition) zentriert sein

(5 P)

b) Betrachten Sie nun den ersten Button im Header, welcher (beim Klicken) zwischen den Aufschriften *Spaltenlayout* und *Reihenlayout* wechselt. Analog wechseln beim `ul` Element mit der id `flexi` die Klassen `columns` und `rows` beim Klicken des Buttons. Implementieren Sie über CSS Selektoren und Flexbox Eigenschaften, dass einmal die Karten in Reihen und einmal in Spalten, abhängig der vorhandenen Klasse, angezeigt werden. Die restlichen Eigenschaften der vorherigen Aufgabe sollen weiter gelten. (2 P)

c) Betrachten Sie nun den zweiten Button im Header, welcher analog zum ersten Button zwischen den Texten *Karten Ordnen* und *Karten mischen* sowie den Klassen `ordered` und `unordered` wechselt. Stellen Sie mittels CSS Selektoren und Flexbox Eigenschaften sicher, dass bei Präsenz der Klasse `unordered` die Karten in der aktuell vorhandenen Reihenfolge angezeigt werden (also wie bisher). Bei Präsenz der Klasse `ordered` sollen die Karten nach ihrer Nummer (Erkennbar im Titel und anhand der ID; letztere ist wohl für Selektoren brauchbarer) geordnet angezeigt werden. *Hinweis: Sie brauchen hierbei wirklich nur die aktuell vorhandenen Karten ordnen und nicht von einer dynamischen Seite ausgehen.* (2 P)